CLAIMS

What is claimed is:

1. A system that facilitates data transformation, comprising:

a component that receives input data; and

a data transformation component that provides a transformation value for the input data derived from, at least in part, at least one unimodular matrix; the transformation value employable to facilitate data protection.

2. The system of claim 1, the data transformation component additionally utilizes at least one secret key to provide the transformation value.

3. The system of claim 1, the data transformation component utilizes single instruction, multiple data (SIMD) processes to generate the transformation value.

4. The system of claim 1, the data transformation component further provides a *d*-semi-universal hash, where the *d*-semi-universal hash possesses a collision probability of two inputs that differ in *d* locations that is substantially near that of colliding with an independently chosen element of a range.

5. The system of claim 4, the *d*-semi-universal hash comprising a 3-semi-universal hash.

6. The system of claim 1, the data transformation component further employs at least one inter-block chaining process that utilizes at least one transformation value from a first input data block in determining at least one transformation value for a second input data block.

7. The system of claim 1, the data transformation component further utilizes at least one transformation value length doubling process.

8. The system of claim 7, the transformation value length doubling process comprising a single-pass computational process.

9. The system of claim 1, additionally comprising:

a transformation value encrypting component that encrypts the transformation value provided by the data transformation component.

10. The system of claim 1, additionally comprising:

a streaming cipher component that employs transformation data generated by the data transformation component as at least a portion of a key utilized to encrypt the input data.

11. The system of claim 1, the unimodular matrix comprising at least one public matrix.

12. The system of claim 1 comprising a reversible system.

13. The system of claim 1, the transformation value comprising a hash value.

14. The system of claim 13, the hash value comprising a hash value with a collision probability for an input data block of length t words with ℓ-bit size defined by:

$$\Pr[H = H'] \leq 2^{-4\ell+20};$$

where the collision probability is taken over a choice of key, H and H' are hash values computed from two distinct inputs, and $t \leq 50$.

15. The system of claim 13, the data transformation component provides the hash value to at least one selected from the group consisting of a data authentication application and a data integrity application.

16. The system of claim 13, the data transformation component further provides control of a length of the hash value.

17. A method for facilitating data transformation, comprising:

receiving input data; and

generating a transformation value for the input data derived from, at least in part, at least one unimodular matrix; the transformation value employable to facilitate data protection.

18. The method of claim 17, generating the transformation value for the input data comprising:

processing the input data into an input block X of length t words with ℓ-bit size and elements $x_1,...,x_t$;

embedding an ℓ-bit input x_i into a 3×3 matrix, B_i, over a ring of integers modulo 2^ℓ by:

$$x_i \mapsto \begin{bmatrix} A_i & v_i \\ 0\,0 & 1 \end{bmatrix} =: B_i;$$

where $v_i = f_i(x_i)$ is a vector with two elements; A_i is a 2×2 matrix with $\det(A_i) = \pm1$; and $f_i(x)$ is a function defined by multiplication of x_i with a random key a_i, where key a_i and input x_i are ℓ bits, $1 \le i \le t$, and a 2ℓ bit result is denoted as a vector of two ℓ-bit numbers; and

computing a product $B = \begin{bmatrix} A & z \\ 0\,0 & 1 \end{bmatrix}$ of the matrices B_i to provide the transformation value in a form of a hash value of $v(X) = (z, \sigma)$; where $\sigma = \sum_{i=1}^{t} v_i$ and z is a two element vector of a third column of matrix B.

19. The method of claim 18, further comprising:

initializing matrix B *via* equation:

$$B := B_0 \cdot \prod_{i=1}^{l} B_i ;$$

where initial matrix B_0 is denoted by:

$$B_0 = \begin{bmatrix} 1 & 0 & \\ 0 & 1 & z_0 \\ 0 & 0 & 1 \end{bmatrix}; \text{ and}$$

where initial vector z_0 is a two-element value for the input block X.

20. The method of claim 18, further comprising:

initializing the hash value, σ, *via* equation:

$$\sigma = \sigma_0 + \sum_{i=1}^{l} v_i ;$$

where σ_0 is an initial value for the input block X.

21. The method of claim 18, the function $f_i(x)$ comprising an invertible modulo $2^{2\ell}$ that is processible in one computational instruction utilizing a 2ℓ-bit result of multiplication of two ℓ-bit quantities.

22. The method of claim 21, the one computational instruction comprising an instruction from a single instruction, multiple data (SIMD) process.

23. The method of claim 18, further comprising:

performing at least one transformation value length doubling process *via* calculation of an independent second transformation value in parallel with a first transformation value by:

setting functions g_i, $i \le t$, $g(x) = b_i \times_* x$, and $u_i = g_i(x_i)$; and

mapping, $X \mapsto u(X)$, with a hash value u utilizing:

$$C_i := \begin{bmatrix} A_i & u_i \\ 0\,0 & 1 \end{bmatrix}, C_0 := \begin{bmatrix} 1 & 0 & u_0 \\ 0 & 1 & \\ 0 & 0 & 1 \end{bmatrix}, C := C_0 \cdot \prod_{i=1}^{t} C_i =: \begin{bmatrix} A & w \\ 0\,0 & 1 \end{bmatrix}; \quad \text{(Eq. 2)}$$

where the second transformation value is derived from keywords b_i, $1 \le i \le t$, which are independent of keywords a_i, $1 \le i \le t$, that are utilized to derive the first transformation value, v is defined as $v = v_0 + \sum_{i=1}^{t} u_i$, and an overall hash is represented by:

$$(v(X), u(X)) = (z, \sigma, w, v).$$

24. The method of claim 18, further comprising:

enciphering the input data with a streaming process that employs, at least in part, a cipher function that incorporates the random key, a_i, and the input, x_i, utilized in generating the transformation value.

25. The method of claim 24, the cipher function comprising:

$$y_i = a_i x_i + b_i;$$

where b_i is a random key and y_i denotes a ciphered output of input x_i.

26. The method of claim 17, further comprising:

encrypting the transformation value.

27. The method of claim 17, the transformation value comprising a *d*-semi-universal hash, where the *d*-semi-universal hash possesses a collision probability of two inputs that differ in *d* locations that is substantially near that of colliding with an independently chosen element of a range.

28. The method of claim 17, further comprising:

employing at least one inter-block chaining process that utilizes at least one transformation value from a first input data block in determining at least one transformation value for a second input data block.

29. The method of claim 17 comprising a reversible method.

30. The method of claim 17, the transformation value comprising a hash value.

31. The method of claim 30, the hash value comprising a hash value with a collision probability for an input data block of length t words with ℓ-bit size defined by:

$$\Pr[H = H'] \leq 2^{-4\ell+20};$$

where the collision probability is taken over a choice of key, H and H' are hash values computed from two distinct inputs, and $t \leq 50$.

32. The method of claim 17, further comprising:

controlling the transformation value length.

33. The method of claim 17 employed in a checksum process.

34. A system that facilitates data transformation, comprising:

means for receiving input data; and

means for providing a data protection transformation value for the input data derived from, at least in part, at least one unimodular matrix.

35. A method for facilitating data transformation, comprising:

receiving an ℓ-bit input x_i of an input data block of length t words, where $1 \leq i \leq t$; and

calculating a block hash by embedding the input x_i into a semidirect product, $G \ltimes H$, *via* mapping the input x_i to $(A_i, f_i(x_i))$; where G represents a group of unimodular matrices over multiplication ($G = SL_2(\mathbb{Z})$), H represents a group of 2-dimensional vectors modulo 2^ℓ over addition ($H = \mathbb{Z}_{2^\ell}^2$), A_i denotes a 2×2 matrix with $\det(A_i) = \pm 1$, and the semidirect product $G \ltimes H$ is defined as a natural homomorphism taking elements of G to automorphisms of H *via* matrix vector products.

36. A data packet, transmitted between two or more computer components, that facilitates data protection, the data packet comprising, at least in part, information relating to a data transformation system that utilizes, at least in part, at least one unimodular matrix to provide a transformation value for input data to facilitate in protection of the input data.

37. A computer readable medium having stored thereon computer executable components of the system of claim 1.

38. A device employing the method of claim 17 comprising at least one selected from the group consisting of a computer, a server, and a handheld electronic device.

39. A device employing the system of claim 1 comprising at least one selected from the group consisting of a computer, a server, and a handheld electronic device.